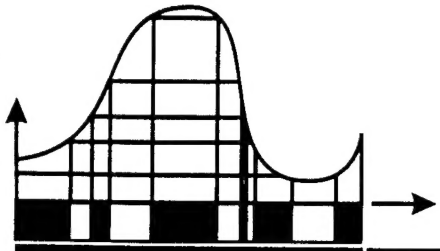

Center for Foundations of Intelligent Systems

Technical Report
98-16

On Explicit Reflection in Theorem Proving and Formal Verification

S. N. ARTEMOV

December 1998

CORNELL
UNIVERSITY

625 Rhodes Hall, Ithaca, NY 14853 (607) 255-8005

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 1 March 1999	3. REPORT TYPE AND DATES COVERED TECHNICAL

4. TITLE AND SUBTITLE
ON EXPLICIT REFLECTION IN THEOREM PROVING AND FORMAL VERIFICATION

5. FUNDING NUMBERS
DAAH04-96-1-0341

6. AUTHOR(S)
S.N. ARTEMOV

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Regents of the University of California
c/o Sponsored Projects Office
336 Sproul Hall
Berkeley, CA 94720-5940

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 35873.133-MA-MUR

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The basic properties of soundness, extensibility, and stability required from a verification system $\mathcal{V}$ taken in full yield the necessity of having a reflection rule in every such $\mathcal{V}$. However, the reflection rule based on the Gödel provability predicate (implicit provability predicate) leads to a "reflection tower" of theories which cannot be formally verified.

The paper introduces an explicit reflection mechanism which can be verified inside the system. This circumvents the reflection tower and provides a strict justification for the verification process. On the practical side, the paper gives specific recommendations concerning the verification of inference rules and building a verifiable reflection mechanism for a theorem proving system.

14. SUBJECT TERMS	15. NUMBER OF PAGES
formal verification, reflection, theorem provers, verification systems. model logic	16
	16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Technical Report
98-16

On Explicit Reflection in Theorem Proving and Formal Verification

S. N. Artemov

December 1998

On explicit reflection in theorem proving and formal verification*

Sergei N. Artemov †

Abstract

The basic properties of soundness, extensibility, and stability required from a verification system $\mathcal{V}$ taken in full yield the necessity of having a reflection rule in every such $\mathcal{V}$. However, the reflection rule based on the Gödel provability predicate (implicit provability predicate) leads to a "reflection tower" of theories which cannot be formally verified.

The paper introduces an explicit reflection mechanism which can be verified inside the system. This circumvents the reflection tower and provides a strict justification for the verification process. On the practical side, the paper gives specific recommendations concerning the verification of inference rules and building a verifiable reflection mechanism for a theorem proving system.

1 Introduction

There is a large variety of theorem provers and proof checkers which can be used for verification (cf. [8], [1], [11]). The mathematical counterparts of those systems range from first order logic (e.g. in **FOL**) and certain fragments of first order arithmetic to higher order logic (**HOL**), the systems with powerful principles sufficient to accommodate most of the classical mathematics (**Mizar**) and most of the computational and constructive tools (**Nuprl**). The underlying logic of such systems can be either classical or intuitionistic.

In this paper we assume that

> *The degree of confidence in a fact verified by a certain system is not higher than the degree of confidence in the system itself.*

*Technical Report CFIS 98-16, Cornell University. Lecture notes of the talk given by the author at the PRL Seminar of the Department of Computer Science, Cornell University, on November 24, 1998.

†Center for Foundation of Intelligent System, Cornell University, email:artemov@math.cornell.edu. The research described in this paper was supported in part by ARO under the MURI program "Integrated Approach to Intelligent Systems", grant DAAH04-96-1-0341, by DARPA under program LPE, project 34145, and by the Russian Foundation for Basic Research, grant 96-01-01395.

This paradigm yields the necessity to keep an account of the tools used in a given verification process. This includes the verification system $\mathcal{V}$ itself along with an exact description of the set of all metamathematical assumptions $\mathcal{M}$ made in the process of verification. Therefore, the set of beliefs which the verification is based upon should include $\mathcal{V} \cup \mathcal{M}$. Without loss of generality we assume in this paper that a metatheory $\mathcal{M}$ of a given verification system $\mathcal{V}$ contains $\mathcal{V}$, therefore, $\mathcal{V} \cup \mathcal{M} = \mathcal{M}$

> For example, suppose we want to verify a statement F by means of the first order arithmetic $\mathcal{PA}$ (i.e. $\mathcal{V} = \mathcal{PA}$). One of the possible ways to put this problem on a formal setting is to say that our goal consists in establishing that $\mathcal{PA} \vdash \mathit{Provable}(F)$, where $\mathit{Provable}(F)$ is a formal statement saying that "F is provable by certain formal tools". Suppose that we have established that $\mathcal{ZF} \vdash \mathit{Provable}(F)$, where $\mathcal{ZF}$ is the Zermelo-Frenkel set theory (a much stronger theory than $\mathcal{PA}$). This corresponds to a realistic situation when a verifier uses the power of all of mathematics, not only the elementary methods formalizable in $\mathcal{PA}$. Here is the sketch of the standard metamathematical argument which under certain assumptions about $\mathcal{ZF}$ concludes that in fact $\mathcal{PA} \vdash \mathit{Provable}(F)$: assume that $\mathcal{ZF}$ is ω-consistent (cf. [14],[7],[15]); since $\mathit{Provable}(F)$ is an arithmetical Σ_1 statement, this yields that $\mathit{Provable}(F)$ is true and, by the Σ_1-completeness of $\mathcal{PA}$, $\mathcal{PA} \vdash \mathit{Provable}(F)$. On the one hand, we have succeeded in establishing that $\mathcal{PA} \vdash \mathit{Provable}(F)$. On the other hand, at the metalevel of this argument we have used the power of $\mathcal{ZF}$ and even the assumption of ω-consistency of $\mathcal{ZF}$. A total account of the beliefs involved in this verification process should include this assumption, which, by the way, has never been and could not possibly be proven by any usual consistent mathematical means[1].

In this paper we will try to demonstrate the following three points:

1. Some form of the reflection rule is a necessary part of an extendable verification system. This will emerge as a natural corollary of the basic soundness, extensibility, and stability assumptions (cf. [8]) about a verification system.

2. The traditional reflection based on the implicit provability predicate cannot be verified in full. It is well-known that even if the implicit reflection is a valid rule in a given system $\mathcal{V}$, its verification cannot be made inside $\mathcal{V}$ (cf. [8], [12], [1], [11]). The present paper demonstrates that the natural metatheory of the "reflection tower" of the implicit reflection rules is not computably enumerable and subsumes all true Π_1-sentences. If one takes into account these hidden metamathematical costs, then within the theory of implicit provability the verification goal of establishing a fact F in $\mathcal{V}$ by formally verifying in $\mathcal{V}$ a proof of F is not achievable.

[1] A better way to present the verification solution from this example would be to simply admit that we are doing the verification in $\mathcal{V} = \mathcal{ZF}$ and thus to restrict the set of beliefs to $\mathcal{ZF}$.

3. There is a new reflection mechanism: "explicit reflection" (introduced in the present paper), which is verifiable in the system itself. The explicit reflection circumvents the reflection tower and provides the strict justification of verification. Explicit reflection requires more information in order to certify the premises of the reflection rule. However, this additional information are usually available in real processes of verification; the old implicit provability model just has not had a mechanism of its utilisation.

On the theoretical side, this paper provides a foundational justification of the verification process. On the practical side, the paper gives specific recommendations concerning the verification of the admissible rules and building a verifiable reflection mechanism for a theorem proving system.

2 Verification systems

2.1 Definition. Under a *verification system* $\mathcal{V}$ we will understand a formal theory satisfying the following conditions a) – d):

a) The underlying logic of $\mathcal{V}$ is either classical or intuitionistic.

b) Proofhood in $\mathcal{V}$ is decidable, therefore theoremhood in $\mathcal{V}$ is computably enumerable. Note that by the well-known Craig Theorem the former follows from the latter for an appropriate choice of axiom system.

c) $\mathcal{V}$ is strong enough to represent any computable function and decidable relation. In particular, given a decidable relation $R(\vec{x})$ one can construct a formula $\mathcal{R}(\vec{x})$ of $\mathcal{V}$ such that for any closed terms $\vec{t}$

$$\text{“}R(\vec{t})\text{” implies } \mathcal{V} \vdash \mathcal{R}(\vec{t}) \text{ and “not } R(\vec{t})\text{” implies } \mathcal{V} \vdash \neg\mathcal{R}(\vec{t}).$$

d) $\mathcal{V}$ has some sort of a numeration of syntax mechanism in the style of [8], [1]. In particular, there is an injective function *rep* which maps syntactic objects like terms, formulas, finite sequences of formulas, sequents, finite trees labeled by sequents, derivation trees, etc., into standard ground terms of $\mathcal{V}$. The usual notation used in this case is $\ulcorner s \urcorner = \mathit{rep}(s)$. The function *rep* and its inverse are both computable. We assume that $\mathcal{V}$ is able to derive formalizations of “usual” combinatory properties of the syntactic objects at a level corresponding to the first order intuitionistic arithmetic $\mathcal{HA}$.

It follows from b) and c) that there is a total computable function which given $R(\vec{t})$ returns a proof of $\mathcal{R}(\vec{t})$ in the former case, and a proof of $\neg\mathcal{R}(\vec{t})$ in the latter case. For the sake of notational simplicity we will use the same names for the informal objects (relations, functions, numbers) and for their formal counterparts (formulas, terms, ground terms) whenever unambiguous.

Examples of verification systems: the first order arithmetic $\mathcal{PA}$; the first order intuitionistic arithmetic $\mathcal{HA}$ and its extensions; second order arithmetic; Martin-Löf type theory $\mathcal{ITT}$; formal set theory $\mathcal{ZF}$; etc. Note that all the above conditions on $\mathcal{V}$ have a purely constructive syntactic character. We have assumed neither semantic properties of $\mathcal{V}$ (e.g. soundness with respect to some semantics) , nor metamathematical ones (consistency, ω-consistency, etc.).

2.2 Definition. For any verification system $\mathcal{V}$ there is a provably decidable (i.e. from Δ_1) formula $Proof(x,y)$ of $\mathcal{V}$ (called *a proof predicate*) obtained by a natural formalization of the inductive definition of derivation in $\mathcal{V}$ (cf. [9], [8], [1]). In particular, $Proof(\ulcorner\mathcal{D}\urcorner,\ulcorner\varphi\urcorner)$ holds iff $\mathcal{D}$ is a proof of φ in $\mathcal{V}$. The Gödel *provability predicate* $Provable(y)$ is defined as $\exists x\, Proof(x,y)$. We will use the notation $\Box\varphi$ for $Provable(\ulcorner\varphi\urcorner)$ and $[\![p]\!]\varphi$ for $Proof(p,\ulcorner\varphi\urcorner)$. For any finite set of $\mathcal{V}$-formulas Γ by $\Box\Gamma$ we mean the conjunction of $\Box\psi$'s for all $\psi\in\Gamma$.

2.3 Definition. The consistency formula $Consis(\mathcal{V})$ is defined as $\neg\Box\bot$, where $\bot$ is the standard *false* formula in $\mathcal{V}$. The informal meaning of $Consis(\mathcal{V})$ is that there is no a proof of *false* in $\mathcal{V}$: this is one of the equivalent formulations of the consistency assertion of $\mathcal{V}$ in the language of $\mathcal{V}$.

We will refer to the provability predicate $\Box(\cdot)$ as the *implicit provability predicate*. The reason for choosing this name lies in the fact that in the formula $\Box\varphi$ (i.e. $\exists x\, Proof(x,\ulcorner\varphi\urcorner)$) the proof is represented implicitly by the existential quantifier, which does not provide any specification of this proof.

The implicit provability predicate has been studied extensively since its invention by Gödel in 1930. The milestone results here are the second Gödel incompleteness theorem (cf. [14], [7])), which states that

$$\textit{If } \mathcal{V} \textit{ is consistent, then } \not\vdash Consis(\mathcal{V}),$$

and the Löb theorem which says that

$$\mathcal{V}\vdash\Box\varphi\to\varphi \textit{ implies } \mathcal{V}\vdash\varphi.$$

By the well-known Hilbert-Bernays lemma (cf. [14],[7]),

$$\mathcal{V}\vdash\varphi \quad \textit{implies} \quad \mathcal{V}\vdash\Box\varphi.$$

This lemma can be considered as a justification of the *formalization rule* $\varphi/\Box\varphi$ for $\mathcal{V}$, which states that every proof in $\mathcal{V}$ can be formalized in $\mathcal{V}$. The proof of the formalization rule is purely syntactic and does not involve any extra assumptions about $\mathcal{V}$. Moreover, this rule can be formalized and proven inside $\mathcal{V}$ (cf. [14], [7]):

$$\mathcal{V}\vdash\Box\varphi\to\Box\Box\varphi.$$

Below we will use one more fact about the provability operator $\Box$, usually attributed to Hilbert, Bernays and Löb (cf.[14],[7]):

$$\mathcal{V} \vdash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi).$$

3 Stability requires reflection

The basic properties required from a verification system are soundness, extensibility, and stability ([8]). We will discuss soundness in Section 4. Extensibility and stability will appear in this section below.

3.1 Definition. A *rule of inference* R in the language of $\mathcal{V}$ is a computable function from a decidable set of finite sets of $\mathcal{V}$-formulas to the set of $\mathcal{V}$-formulas. The usual notation for a rule of inference R is Γ/φ, where Γ indicates the argument of R (premises), and φ the value $R(\Gamma)$ of R (conclusion). For the sake of notational convenience we will not distinguish between a finite set of formulas Γ and one formula which is the conjunction of all formulas from Γ when unambiguous. We would like to think that such an abuse of notation will be tolerated by a reader.

3.2 Definition. A rule of inference Γ/φ is *derived in* $\mathcal{V}$ if $\mathcal{V} \vdash \Gamma \to \varphi$.
A rule of inference Γ/φ is *implicitly verified in* $\mathcal{V}$ if $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$.
A rule Γ/φ is *admissible in* $\mathcal{V}$ if $\mathcal{V} \vdash \Gamma$ implies $\mathcal{V} \vdash \varphi$.

3.3 Lemma.
1. *Every derived rule is implicitly verified, but not vice versa.*
2. *Every derived rule is admissible, but not vice versa.*

Proof. 1. Let $\mathcal{V} \vdash \Gamma \to \varphi$. By the formalization rule, $\mathcal{V} \vdash \Box(\Gamma \to \varphi)$. By the properties of provability operator (Section 2), $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$. Here are examples of implicitly verified rules that are not derivable: $\varphi/\forall x\varphi$ (*generalization*), $\varphi/\Box\varphi$ (*formalization*), $\Box\varphi \to \varphi/\varphi$ (*Löb's rule*), $\neg\neg\sigma)/\sigma$, where σ is a Σ_1 sentence (*Markov rule* for intuitionistic arithmetic $\mathcal{HA}$, cf. [16]).

2. If $\mathcal{V} \vdash \Gamma \to \varphi$ and $\mathcal{V} \vdash \Gamma$, then $\mathcal{V} \vdash \varphi$. The rules *generalization*, *formalization*, *Löb's rule*, *Markov rule* from above are all admissible but not derived.
◀

The extensibility property of $\mathcal{V}$ is understood ([8]) as a technical possibility to extend $\mathcal{V}$ by adding rules of inference verified in $\mathcal{V}$. We accept the understanding of stability as conservativity of extensions by implicitly verified rules (cf. [1], [11]).

3.4 Definition. System $\mathcal{V}' \supseteq \mathcal{V}$ is *conservative* over $\mathcal{V}$ if for any formula ψ

$$\mathcal{V}' \vdash \psi \text{ implies } \mathcal{V} \vdash \psi.$$

A system $\mathcal{V}$ is *implicitly stable* if for any rule Γ/φ implicitly verified in $\mathcal{V}$ the system $\mathcal{V}+\Gamma/\varphi$ is conservative over $\mathcal{V}$.

3.5 Definition. The *implicit reflection rule* $IRR(\mathcal{V})$ is the rule $\Box\varphi/\varphi$ where $\Box\varphi$ represents the provability of φ in $\mathcal{V}$.

3.6 Example. Here is the standard example of a formal theory for which the implicit reflection rule does not hold ([9]): $\mathcal{V} = \mathcal{PA} + \neg Consis(\mathcal{PA})$. This system is consistent, i.e. $\mathcal{V} \nvdash \bot$. On the other hand $\mathcal{V} \vdash \Box\bot$, where $\Box$ stands for provability in this particular $\mathcal{V}$.

3.7 Theorem. *A verification system $\mathcal{V}$ is implicitly stable iff the implicit reflection rule $IRR(\mathcal{V})$ is admissible in $\mathcal{V}$.*

Proof. Let $\mathcal{V}$ be an implicitly stable system. Let us consider the rule R_φ consisiting of a single pair $(TRUE, \varphi)$, where $TRUE$ is the propositional constant for true statements in $\mathcal{V}$. Since $\mathcal{V} \vdash TRUE$, we also have $\mathcal{V} \vdash \Box(TRUE)$. By implicit stability of $\mathcal{V}$, for all φ, ψ if $\mathcal{V} \vdash \Box(TRUE) \to \Box\varphi$ and $\mathcal{V} + TRUE/\varphi \vdash \psi$, then $\mathcal{V} \vdash \psi$. Equivalently, for all φ, ψ if $\mathcal{V} \vdash \Box\varphi$ and $\mathcal{V} + \varphi \vdash \psi$, then $\mathcal{V} \vdash \psi$. Let ψ be φ. Then $\mathcal{V} \vdash \Box\varphi$ implies $\mathcal{V} \vdash \psi$ for all φ, therefore $IRR(\mathcal{V})$ is admissible in $\mathcal{V}$.

Let now $IRR(\mathcal{V})$ be admissible in $\mathcal{V}$, i.e. $\mathcal{V} \vdash \Box\varphi$ implies $\mathcal{V} \vdash \varphi$, and let Γ/φ be an implicitly verified rule, i.e. $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$. By an induction on the derivation in $\mathcal{V}+\Gamma/\varphi$ we prove that $\mathcal{V}+\Gamma/\varphi \vdash \psi$ implies $\mathcal{V} \vdash \psi$. The induction basis holds because $\mathcal{V}$ and $\mathcal{V}+\Gamma/\varphi$ have the same set of axioms. The induction step in the case of a rule other than Γ/φ is trivial. Let ψ be obtained in $\mathcal{V}+\Gamma/\varphi$ by the rule Γ/φ, i.e. there is specific Γ_1 such that Γ_1/ψ is a special case of the rule Γ/φ and $\mathcal{V}+\Gamma/\varphi \vdash \Gamma_1$. By the induction hypothesis, $\mathcal{V} \vdash \Gamma_1$. By the formalization rule in $\mathcal{V}$, $\mathcal{V} \vdash \Box\Gamma_1$. Since the rule Γ/φ is implicitly verified, we have $\mathcal{V} \vdash \Box\Gamma_1 \to \Box\psi$, therefore $\mathcal{V} \vdash \Box\psi$. By the rule $IRR(\mathcal{V})$, $\mathcal{V} \vdash \psi$.

◀

Extensibility by derived rules however, can be verified inside the system without any additional assumptions.

3.8 Theorem. *An extension of a verification system $\mathcal{V}$ by a derived rule is provably in $\mathcal{V}$ conservative.*

Proof. The following argument can be formalized in $\mathcal{V}$. Let $\mathcal{V} \vdash \Gamma \to \varphi$ and $\mathcal{V}' = \mathcal{V} + \Gamma/\varphi$. By the induction on a proof in $\mathcal{V}'$ similar to the one from the proof of Theorem 3.7 we show that for any formula ψ if $\mathcal{V}' \vdash \psi$, then $\mathcal{V} \vdash \psi$. We consider the most important case in the induction step. Let ψ be obtained in $\mathcal{V} + \Gamma/\varphi$ by the rule Γ/φ, i.e. there is specific Γ_1 such that Γ_1/ψ is a special case of the rule Γ/φ and $\mathcal{V} + \Gamma/\varphi \vdash \Gamma_1$. By the induction hypothesis, $\mathcal{V} \vdash \Gamma_1$. Since $\mathcal{V} \vdash \Gamma_1 \to \psi$, we got $\mathcal{V} \vdash \psi$.
◀

3.9 Comment. **Nuprl** has the mechanism of *tactics* based on the extension by the derived rules. As we see from 3.8, this mechanism can be justified inside the system as does not need any additional assumptions. Although correct this mechanism is not as general as the extensions by verified rules (cf. Lemma 3.3(2)).

4 Metamathematical cost of soundness and implicit stability

In this section we will find lower and upper bounds for the minimal metatheory $\mathcal{M}$ capable of establishing soundness and stability of a given verification system $\mathcal{V}$.

We will use the Turing progression as the standard scale to measure the metamathematical strength of a given extension of the basic theory ([13]). The Turing progression $\mathcal{V}^c_\alpha$ of theories (cf. [17], [10], [2]) for $\mathcal{V}$ is obtained from $\mathcal{V}$ by iterating the consistency assumptions along the Church-Kleene system of constructive ordinals α.

We consider the first ω theories from the Turing progression.

$$\mathcal{V}^c_0 = \mathcal{V}, \qquad \mathcal{V}^c_{n+1} = \mathcal{V}^c_n + \mathit{Consis}(\mathcal{V}^c_n), \qquad \mathcal{V}^c_\omega = \bigcup_n \mathcal{V}^c_n.$$

If $\mathcal{V}$ is correct with respect to the standard model of arithmetic, then the following strict inclusions hold:

$$\mathcal{V}^c_0 \subset \mathcal{V}^c_1 \subset \mathcal{V}^c_2 \subset \ldots \subset \mathcal{V}^c_\omega.$$

Soundness was described in [8] as the condition that "We must be entirely convinced that any proof of a theorem which the system certifies as correct should indeed be so." A straightforward way to formalize soundness would be to assume some sort of the semantics for $\mathcal{V}$, to take $\mathcal{M}$ powerful enough to express the notion of truth for the $\mathcal{V}$-formulas and to establish inside $\mathcal{M}$ a formal analogue of the statement

for every sentence φ if φ is provable then φ is true.

This approach would require a fairly strong $\mathcal{M}$. In particular, one needs to extend the language of $\mathcal{V}$ in order to write down formulas *"φ is true"*; by the well-known Tarski theorem there is no such formula in the language of $\mathcal{V}$ itself.

In fact, soundness of a verification system $\mathcal{V}$ deals with the true values of formal statements of an especially simple type, namely provable Δ_1 sentences $[\![t]\!]\varphi$.

4.1 Theorem.

1. The following conditions are equivalent: a) $\mathcal{V} \vdash [\![t]\!]\varphi$ implies $[\![t]\!]\varphi$ is true; b) $\mathcal{V}$ is consistent.

2. $\mathcal{V}$ suffices to establish 1.

Proof. If 1a then the false sentences of the kind $[\![t]\!]\varphi$ are not provable in $\mathcal{V}$, therefore $\mathcal{V}$ is consistent. Suppose 1b and let $\mathcal{V} \vdash [\![t]\!]\varphi$. If $[\![t]\!]\varphi$ were false, then $\mathcal{V} \vdash \neg[\![t]\!]\varphi$, by Δ_1 completeness of $\mathcal{V}$. This leads to a contradiction in $\mathcal{V}$.

2. The straightforward formalization of the proof of 1 with the use of provable Δ_1 completeness of $\mathcal{V}$.

◀

4.2 Corollary. Simple consistency of $\mathcal{V}$ is necessary and sufficient for soundness of a verification system $\mathcal{V}$.

Now we will figure out what metatheory can establish implicit stability.

4.3 Definition. by "**$\mathcal{V}$ is stable**" we understand the $\mathcal{V}$-formula which is the natural formalization of the stability property of $\mathcal{V}$. By "**implicit reflection rule is admissible in $\mathcal{V}$**", or equivalently

$$\forall x(\Box\Box x \to \Box x),$$

we mean the natural formalization in the language of $\mathcal{V}$ of the property that $IRR(\mathcal{V})$ is admissible in $\mathcal{V}$.

4.4 Theorem. $\mathcal{V} \vdash$ "**$\mathcal{V}$ is stable**" $\iff$ "**implicit reflection rule is admissible in $\mathcal{V}$**"

Proof. The straightforward (though delicate) formalization of the proof of Theorem 3.7.

◀

4.5 Theorem. *Implicit stability of an ω-consistent verification system is not provable in this system.*

Proof. By Theorem 4.4, implicit stability is provable in $\mathcal{V}$ iff $\mathcal{V} \vdash \forall x(\Box\Box x \to \Box x)$. Let x is the code of $\bot$. Then $\mathcal{V} \vdash \Box\Box\bot \to \Box\bot$. By Löb's theorem, $\mathcal{V} \vdash \Box\bot$, which is impossible for an ω-consistent $\mathcal{V}$.
◀

It follows from the above that the minimal metatheory for soundness and implicit stability is

$$\mathcal{M} = \mathcal{V} + \mathit{Consis}(\mathcal{V}) + \forall x(\Box\Box x \to \Box x).$$

4.6 Theorem. *If $\mathcal{V}$ is correct with respect to the standard model of arithmetic then the metatheory for soundness and implicit stability strictly subsumes the first ω steps of the Turing progression.*

Proof. In order to establish $\mathcal{V}^c_\omega \subset \mathcal{M}$ consider the formulas $\Box^0\bot = \bot$, $\Box^{n+1}\bot = \Box(\Box^n\bot)$. First of all we note that under the assumptions made about $\mathcal{V}$ the formula $\mathit{Consis}(\mathcal{V}^c_n)$ is provably equivalent in $\mathcal{V}$ to $\neg\Box^{n+1}\bot$ (cf. [2]). Indeed, $\mathit{Consis}(\mathcal{V}^c_0)$ is $\mathit{Consis}(\mathcal{V})$, i.e. $\neg\Box\bot$. Then $\mathit{Consis}(\mathcal{V}^c_1)$ is a formula stating that $\mathcal{V} + \mathit{Consis}(\mathcal{V}) \nvdash \bot$, i.e. $\mathcal{V} + \neg\Box\bot \nvdash \bot$. This is equivalent to $\mathcal{V} \nvdash \neg\Box\bot \to \bot$ and $\mathcal{V} \nvdash \Box\bot$. Therefore, $\mathit{Consis}(\mathcal{V}^c_1)$ is equivalent to $\neg\Box\Box\bot$. Similar argument works for $n = 2, 3, 4, \ldots$.

Now we show how to derive all $\neg\Box^n\bot$, $n = 1, 2, 3, \ldots$ in $\mathcal{M}$. The case $n = 1$ is covered by the assumption that $\mathcal{M} \vdash \mathit{Consis}(\mathcal{V})$, which is equivalent to $\mathcal{M} \vdash \neg\Box\bot$, or $\mathcal{M} \vdash \Box\bot \to \bot$ For $n = 2$ put $x = \bot$ in $\forall x(\Box\Box x \to \Box x)$. Then $\mathcal{M} \vdash \Box\Box\bot \to \Box\bot$. Since we have already had $\mathcal{M} \vdash \Box\bot \to \bot$, we conclude that $\mathcal{M} \vdash \Box\Box\bot \to \bot$, i.e. $\mathcal{M} \vdash \neg\Box\Box\bot$. A similar argument works for $n = 3, 4, 5, \ldots$. Thus

$$\mathcal{V}^c_\omega \subset \mathcal{M}.$$

Now we will check that $\mathcal{V}^c_\omega \neq \mathcal{M}$. Suppose

$$\mathcal{V}^c_\omega \vdash \mathit{Consis}(\mathcal{V}) \wedge \forall x(\Box\Box x \to \Box x).$$

By the compactness argument, there is a natural number n such that

$$\mathcal{V}^c_n \vdash \mathit{Consis}(\mathcal{V}) \wedge \forall x(\Box\Box x \to \Box x).$$

Since $\mathcal{V}^c_\omega \subset \mathcal{M}$, $\mathcal{M}$ proves the consistency of $\mathcal{V}^c_n$. Therefore

$$\mathcal{V}^c_n \vdash \mathit{Consis}(\mathcal{V}^c_n),$$

which is impossible by the second Gödel incompleteness theorem for $\mathcal{V}^c_n$.
◀

5 Metamathematical cost of implicit reflection

In an ω-consistent verification system $\mathcal{V}$ the rule of implicit reflection $IRR(\mathcal{V})$ is admissible, i.e. $\mathcal{V} \vdash \Box\varphi$ yields $\mathcal{V} \vdash \varphi$ for any formula φ. The most simple formalization of the admissibility property is the scheme $\Box\Box\varphi \rightarrow \Box\varphi$, where $\Box\psi$ stands for the formula of provability of ψ in $\mathcal{V}$. A general procedure of incorporating implicit reflection rule into a verification system $\mathcal{V}$ may be presented by the following *reflection tower* of extensions of $\mathcal{V}$ (cf. [12], [1], [11]):

$$\mathcal{V}^r_0 = \mathcal{V}, \quad \mathcal{V}^r_{\alpha+1} = \mathcal{V}^r_\alpha + IRR(\mathcal{V}^r_\alpha), \quad \mathcal{V}^r_\gamma = \bigcup_{\beta \prec \gamma} \mathcal{V}^r_\beta \text{ for a limit ordinal } \gamma.$$

For the sake of simplicity we assume in this section that $\mathcal{V}$ is sound with respect to the standard model of arithmetic.

In this section we will try to figure out what natural metatheory is able to establish the admissibility of all the reflection rules from the reflection tower.

5.1 Definition. *Implicit reflection principle* $IRP(\mathcal{V})$ for a given system $\mathcal{V}$ is the scheme of formulas

$$\{\Box\varphi \rightarrow \varphi \mid \varphi \text{ is a sentence of } \mathcal{V}\}.$$

Let us consider *Feferman's progression* of extensions of $\mathcal{V}$ by the implicit reflection principles ([10]):

$$\mathcal{V}^p_0 = \mathcal{V}, \quad \mathcal{V}^p_{\alpha+1} = \mathcal{V}^p_\alpha + IRP(\mathcal{V}^p_\alpha), \quad \mathcal{V}^p_\gamma = \bigcup_{\beta \prec \gamma} \mathcal{V}^p_\beta \text{ for a limit ordinal } \gamma.$$

The system $\mathcal{V}^p_1$ proves admissibility of implicit reflection in $\mathcal{V}^r_0$, i.e. the scheme of formulas $\Box\Box\varphi \rightarrow \Box\varphi$. In addition $\mathcal{V}^p_1 \subset \mathcal{V}^r_1$, since every instance of the rule $\Box\varphi/\varphi$ in a proof in $\mathcal{V}^r_1$ can be emulated by the axiom $\Box\varphi \rightarrow \varphi$. Moreover, the inclusion $\mathcal{V}^p_1 \subset \mathcal{V}^r_1$ can be established in $\mathcal{V}$. Iterating this argument one can show that $\mathcal{V}^p_{\alpha+1}$ is the theory capable of establishing admissibility of the implicit reflection rule for $\mathcal{V}^r_\alpha$.

How bad really is the reflection tower for $\mathcal{V}$? The natural metatheory capable of verifying the whole reflection tower is the limit of Feferman's progression $\mathcal{V}^p_\alpha$ for all constructive ordinals α.

5.2 Proposition. ([10]) *The limit of* $\mathcal{V}^p_\alpha$ *for all constructive ordinals* α *equals*

$$\mathcal{V} + \textit{all true } \Pi_1\textit{-sentences.}$$

It follows from the above that the natural metatheory for the reflection tower is not computably enumerable, and could not possibly be verified by any sound mathematical means. It

contains, for example, the consistency statements for all consistent axiomatic theories, among them $Consis(\mathcal{ZF})$ (provided $\mathcal{ZF}$ is consistent).

In the next section we describe explicit reflection, which is verifiable by means of the system itself and thus circumvents the reflection tower.

6 Explicit reflection for verification systems

An alternative way to represent provability in a logical setting has been developed in [3] – [6]. The key idea of this approach is to represent provability by a certain family of *proof operators* $[\![t]\!]\varphi$ (i.e. $Proof(t, \ulcorner\varphi\urcorner)$) with an appropriate set of ground proof terms t). As it was shown in [5] and [6], every propositional property of the provability operator can be represented by the family of proof operators with a certain class of finitely generated terms. It is easy to notice that the following *explicit formalization theorem* holds: For every sentence φ such that $\mathcal{V} \vdash \varphi$ there is a ground term t of $\mathcal{V}$ such that $\mathcal{V} \vdash [\![t]\!]\varphi$.

6.1 Definition. The *explicit reflection principle* $ERP(\mathcal{V})$ is the scheme of formulas $[\![t]\!]\varphi \to \varphi$ for all sentences φ and all ground terms t.

6.2 Lemma. (Derivability of explicit reflection [3]). *For any ground term t and formula φ*

$$\mathcal{V} \vdash [\![t]\!]\varphi \to \varphi.$$

Proof. We give a constructive proof of this lemma which delivers an algorithm for constructing a derivation of $[\![t]\!]\varphi \to \varphi$ in $\mathcal{V}$ given φ and t. First of all, by the proof checking procedure we calculate the truth value of $[\![t]\!]\varphi$. If this value is *TRUE*, then the ground term t represents a derivation of φ, from which by a straightforward reconstruction, we obtain the proof of $[\![t]\!]\varphi \to \varphi$. If the proof checker on $[\![t]\!]\varphi$ returns *FALSE*, then by the corresponding procedure mentioned in 2.1, we get the proof of $\neg[\![t]\!]\varphi$ in $\mathcal{V}$. From that by the straightforward transformation, we get the proof of $[\![t]\!]\varphi \to \varphi$.
◀

6.3 Corollary. *There is an algorithm which given a formula φ and a ground term t returns the ground term p such that*

$$\mathcal{V} \vdash [\![p]\!]([\![t]\!]\varphi \to \varphi).$$

6.4 Definition. The *explicit reflection rule* $ERR(\mathcal{V})$ is the rule $[\![t]\!]\varphi / \varphi$ for all ground terms t and all sentences φ.

6.5 Definition. A rule Γ/φ is *explicitly verifiable in* $\mathcal{V}$ if there is a total computable function f such that $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$.

It is clear from the definitions that "explicitly verifiable" implies "implicitly verifiable".

6.6 Theorem. *The explicit reflection rule $ERR(\mathcal{V})$ is explicitly verifiable in $\mathcal{V}$.*

Proof. Let "$\cdot$" be a total and computable "application" function on proof codes, specified by the condition

$$\mathcal{V} \vdash [\![x]\!](\varphi \to \psi) \to ([\![y]\!]\varphi \to [\![x \cdot y]\!]\psi)$$

(cf. [5], [6]). By 6.3, $\mathcal{V} \vdash [\![p]\!]([\![t]\!]\varphi \to \varphi)$ for some ground term p. Therefore,

$$\mathcal{V} \vdash ([\![y]\!][\![t]\!]\varphi \to [\![p \cdot y]\!]\varphi).$$

◀

6.7 Corollary. *The explicit reflection rule $ERR(\mathcal{V})$ is admissible for every verification system $\mathcal{V}$.*

6.8 Definition. A rule of inference included in the description of a system $\mathcal{V}$ is called an *internal rule* of $\mathcal{V}$.

6.9 Lemma. *Every internal rule is explicitly verifiable.*

Proof. There is a straightforward function behind every internal rule Δ/ψ which calculates the code of a proof of ψ given the codes of proofs of Δ. A natural formalization of this function in $\mathcal{V}$ gives a term f such that $\mathcal{V} \vdash [\![y]\!]\Delta \to [\![f(y)]\!]\psi$.
◀

6.10 Definition. An extension $\mathcal{V}'$ of $\mathcal{V}$ is *verifiably equivalent* to $\mathcal{V}$ if there is a computable function g of $\mathcal{V}$ such that $\mathcal{V} \vdash [\![x]\!]'\psi \to [\![g(x)]\!]\psi$, where $[\![x]\!]'\psi$ stands for the formula "x is a proof of ψ in $\mathcal{V}'$. In other words, for a verifiably equivalent extension $\mathcal{V}'$ there is an algorithm that transforms proofs in $\mathcal{V}'$ into proofs of the same facts in $\mathcal{V}$.

6.11 Theorem. *An extension of a verification system by an explicitly verified rule is verifiably equivalent to the original system.*

Proof. Let a rule Γ/φ be explicitly verifiable in a verification system $\mathcal{V}$, i.e. there is a computable function f such that $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$. Let $\mathcal{V}'$ be $\mathcal{V} + \Gamma/\varphi$. The function $g(x)$ works as follows. It travels along the proof tree in $\mathcal{V}'$ coded by x and calculates the code of a proof tree in $\mathcal{V}$ of the same sentence (sequent). If the observed node is a leaf node, then it corresponds to an axiom of $\mathcal{V}'$, which is an axiom of $\mathcal{V}$ as well. In this situation g does not change the the proof at all.

Let the observed node correspond to an application of an internal rule Δ/θ, and let $\vec{u}$ be the values of g on the predecessors of the current node, i.e. $\mathcal{V} \vdash [\![\vec{u}]\!]\Delta$. By lemma 6.9, there is a computable function h such that $\mathcal{V} \vdash [\![\vec{y}]\!]\Delta \to [\![h(\vec{y})]\!]\theta$. Substituting u for y we derive $[\![h(\vec{u})]\!]\theta$ in $\mathcal{V}$. Let g map the observed node to $h(\vec{u})$.

Let the observed node correspond to an application of the new rule Γ/φ, and let $\vec{v}$ be the values of g on the predecessors of this node, i.e. $\mathcal{V} \vdash [\![\vec{v}]\!]\Gamma$. By the conditions of the theorem $\mathcal{V} \vdash [\![\vec{y}]\!]\Gamma \to [\![f(\vec{y})]\!]\varphi$. Substitute v's for y's, conclude that $\mathcal{V} \vdash [\![f(\vec{v})]\!]\varphi$ and let g map the observed node to $f(\vec{v})$.

Eventually, at the root node of the $\mathcal{V}'$-proof (coded by) x the function g returns the code of a $\mathcal{V}$-proof of the formula (sequent) previously proven by x.

◀

7 Practical suggestions

As one can see, explicit reflection avoids some of the troubles inherent in implicit reflection. Here is the list of practical suggestions for the designers of verification systems. Explicit reflection says nothing new for nonextendable systems without reflection mechanism. In such a system the explicit reflection rule has already been used by default when one concludes that $\mathcal{V}$ has verified a fact φ given that $\mathcal{V} \vdash [\![t]\!]\varphi$ for some proof code t.

There are two classes of systems where explicit reflection can bring a significant improvement.

1. **Verification systems with extensibility but without special built-in reflection mechanisms.** Here the use of explicit reflection may be twofold. Firstly, it appears in the *assertion insertion mode* (cf. [8]), when it is established that $\mathcal{V} \vdash [\![t]\!]\varphi$ and then φ is stored as a verified fact (i.e. a new axiom) of $\mathcal{V}$. We have nothing specific to add here, since this mode as presented above (and in [8]) already agrees with the explicit reflection recommendations. Secondly, the explicit reflection appears in the *rule insertion mode*, when Γ/φ is verified in $\mathcal{V}$ and then added to $\mathcal{V}$ as a new inference rule. The explicit reflection suggests verifying the rule Γ/φ in $\mathcal{V}$ explicitly, i.e. by constructing a computable function f such that $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$. By doing this we guarantee that the resulting extension is verified in the old system without any hidden metaassumptions.

If the rule insertion mode uses explicit verification only, then there is no need to

have a special built-in reflection mechanism: provable stability of the system is preserved by explicit verification (Theorem 6.11).

Interestingly enough, there are substantial classes of verification systems where the implicit verification in a certain sense yields the explicit one. For example, in traditional intuitionistic systems $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$ implies $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$ for some computable function f (cf. [16]). However, the proof of this fact itself cannot be formalized in $\mathcal{V}$ and its use in the rule insertion mode leads to some sort of a reflection tower. Therefore, ever for the constructive systems the practical suggestion is to use the explicit verification, i.e. to establish $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$ directly rather than to prove $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$ and then to apply a general theorem of obtaining the explicit verification from the implicit one; this involves some hidden and potentially high metamathematical costs.

2. **Advanced systems with built-in reflection mechanisms.** There is a number of systems which have or intend to have such mechanisms. The paper [11] mentions several of them: **FOL**, **NQTHM**, **HOL** and **Nuprl**. At least one more is coming: **MetaPrl** at Cornell University. Probably more systems will join this set since reflection arguments are surprisingly often used in mathematical and common reasoning. The existing implicit reflection mechanisms in these systems lead to unnecessary metamathematical costs (cf. Section 5). For such systems the idea of having explicit reflection (perhaps, along with the implicit one) might be seriously considered, because the explicit reflection can be added to a system without any extra metamathematical assumptions at all (Theorem 6.6).

Right now within the **Nuprl** research group at Cornell University we are exploring the possibility to build explicit reflection mechanisms in the new generation of **Nuprl** systems.

8 Acknowledgements

I am indebted to Robert Constable for attracting my attention to the problem of reflection in verification systems and for support during my work on this paper. I am also grateful to Stewart Allen, Anil Nerode, Elena Nogina and Vaughan Pratt for valuable suggestions and criticism. Many thanks to Martin Davis for fruitful discussion and for sending me a copy of his paper.

References

[1] S. Allen, R. Constable, D. Howe, and W. Aitken, "The Semantics of Reflected Proofs." In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science*, Los Alamitos, CA, USA, IEEE Computer Society Press, pp. 95-107, 1990.

[2] S. Artemov, *Extensions of theories by the reflection principles and the corresponding modal logics*, Ph.D. Thesis, Moscow, 1979.

[3] S. Artemov and T. Strassen. "The Basic Logic of Proofs," *Lecture Notes in Computer Science*, v. 702, pp. 14-28, Springer-Verlag, 1993.

[4] S. Artemov, "Logic of Proofs," *Annals of Pure and Applied Logic*, v. 67, pp. 29-59, 1994.

[5] S. Artemov. "Operational modal logic," *Technical Report MSI 95-29*, Cornell University, 1995.

[6] S. Artemov, "Explicit provability: the intended semantics for intuitionistic and modal logic" *Technical Report CFIS 98-10*, Cornell University, September 1998.

[7] G. Boolos, *The Unprovability of Consistency: An Essay in Modal Logic*, Cambridge University Press, 1979.

[8] M. Davis and J. Schwartz, "Metamathematical extensibility for theorem verifiers and proof checkers," *Computers and Mathematics with Applications*, v. 5, pp. 217-230, 1979.

[9] S. Feferman, "Arithmetization of metamathematics in a general setting," *Fundamenta Mathematicae*, v. 49, pp. 35-92, 1960.

[10] S. Feferman, "Transfinite recursive progressions of axiomatic theories," *Journal of Symbolic Logic*, v. 27, pp. 259-316, 1962.

[11] J. Harrison, "Metatheory and Reflection in Theorem Proving: A Survey and Critique," University of Cambridge, http://www.dcs.glasgow.ac.uk/ tfm/hol-bib.html#H, 1995.

[12] T. Knoblock and R. Constable, "Formalized metareasoning in type theory," in *Proceedings of the First Annual Symposium on Logic in Computer Science*, Cambridge, MA, USA, IEEE Computer Society Press, pp. 237-248, 1986.

[13] G. Kreisel and A. Levy, "Reflection principles and their use for establishing the complexity of axiomatic systems," *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, v. 14, pp. 97-142, 1968.

[14] C. Smorynski, "The incompleteness theorems," in: *Handbook of Mathematical Logic*, J. Barwise, ed., vol. 4, North-Holland, Amsterdam, pp. 821-865, 1977.

[15] C. Smorynski, *Self-Reference and Modal Logic*, Springer-Verlag, Berlin, 1985

[16] A.S. Troelstra and D. van Dalen, *Constructivism in Mathematics. An Introduction*, v. 1, Amsterdam; North Holland, 1988.

[17] A. Turing, "Systems of logics based on ordinals," *Proceedings of the London Mathematical Society*, ser. 2, v. 45, pp. 161-228, 1939.